

SPIDER Koerier

Maart 2010
Nummer 2010-1

Lente Editie 2010

www.st-SPIDER.nl

Redactioneel

Na een echte Hollandse winter is dan nu de lente aangebroken. Ik hoop dat je nu de SPIDER Koerier lekker in de voorjaarszon kunt gaan lezen, al dan niet onder het genot van een glas naar je voorkeur. We vertrouwen erop dat ook deze keer de verhalen uitdagend genoeg zijn om van te genieten. Het is onze bedoeling dat je bij het lezen ook uitgedaagd wordt om te reageren, of om je eigen ervaringen een keer te delen met de SPIDER leden. De Koerier staat open voor goede verhalen, of het nu een kort of een uitgebreid verslag of artikel is, ik nodig je van harte uit.

De werkgroepen Requirements en P-CMM doen verslag in deze Koerier. We hebben ook een aankondiging van alweer een nieuw CMMI boek. En we hebben de hand weten te leggen op een interessant artikel over het automatiseren van Requirements. Een SPIDER lid geeft zijn ervaring op het gebied van service oriëntatie weer. Uiteraard ontbreekt een introductie van onze voorzitter niet en je kunt alvast lezen wie er op de aanstaande plenaire sessie op 30 maart in Den Bosch zullen komen spreken. Kortom, veel leesplezier.

René Krikhaar

Inhoudsopgave

Redactioneel	1
Inhoudsopgave	1
Van de voorzitter SPIDER	2
Plenaire Sessie: Onderhoudbaarheid	3
SPIDER Werkgroep Requirements	4
Tijd voor verandering	6
PA SMART Requirements, 100% SMART Requirements...finally!!	7
SPIDER Werkgroep P-CMM	12

De activiteiten van SPIDER worden gesponsord door:

	
www.Philips.com	www.sig.eu/nl
	
www.Sogeti.nl	www.dnv.nl

Nieuws van de Q Society	13
Harvesting service orientation	14
De kleine CMMI trilogie compleet	18
Publicatie CMMI voor Ontwikkeling	19
De SPIder Organisatie	20
Opstarten nieuwe Werkgroep SPIder	20
Colofon	21
Nieuwsberichten & evenementenkalender	22

Van de voorzitter SPIder

2009 was voor de Stichting SPIder een succesvol jaar. Als eerste hebben we de stijgende lijn in deelnemers aan de plenaire sessie ook in 2009 kunnen doorzetten. De minst bezochte sessie was de sessie van 25 november in Eindhoven over Agile Development. Alhoewel het de minst bezochte was, werd deze plenaire sessie door 66 (!) leden bezocht. De meeting van 4 maart over proces verbeteren werd door 77 leden bezocht.

Het onderwerp Verandermanagement, van 16 september met Pat O'toole, werd door 82 leden bezocht en de sessie van 13 mei over codekwaliteit werd het best bezocht. In totaal kwamen hier 96 leden op af. Een absoluut record!

Het tweede positieve punt van 2009 dat ik wil benadrukken is onze (betaalde) conferentie. Ondanks de crisis is deze conferentie door 98 deelnemers bezocht.

Daarnaast hebben inmiddels drie van de hoofd sponsors aangegeven om het sponsorschap ook in 2010 te willen continueren. We heten dit jaar SIG, Software Improvement Group, welkom als nieuwe sponsor van SPIder! SIG is een onafhankelijk toezichthouder van strategische automatiseringsprojecten en rapporteert en adviseert hoger management.

Het succes van de plenaire sessies en de conferentie en de waardering van onze sponsors, zijn voor het bestuur signalen dat we op de goede weg zijn. Dat wil echter niet zeggen dat we geen verbetermogelijkheden zien.

Als bestuur hebben we een ambitie geformuleerd waar we in 2012 als SPIder willen staan. Deze ambitie is samen te vatten in drie woorden; Verdiepen, Verbinden en Verbreden.

Verdiepen

Het blijven streven naar verdere kwaliteitsverbetering en professionalisering. De Deming Circle als het ware op onszelf toepassen en kritisch blijven op de kwaliteit van geschreven (Koerier) of gesproken (conferentie, plenaire sessies) bijdragen. Daarbij vasthouden dat we een laagdrempelig platform aanbieden - voor junior schrijvers en sprekers - om hun ideeën en ervaringen aan een breder publiek toegankelijk te maken.

Verbinden

De Stichting SPIder heeft een kleine 1400 leden. Leden die werkzaam zijn in het vakgebied, een eigen netwerk hebben in SPI en kwaliteit en geïnteresseerd zijn om kennis op te doen betreffende de onderwerpen die SPIder op de agenda heeft staan. Van onze 1400 leden is een kwart actief. D.w.z. dat zij plenaire sessies en de conferentie bezoeken, deelnemen in werkgroepen en/of artikelen schrijven voor de

Koerier. Wij willen graag de betrokkenheid van onze leden vergroten om daarmee de kennisuitwisseling te vergroten.

Verbreden

Verbreiding willen we bereiken op twee vlakken. Ten eerste willen we onze samenwerking verbreden met verenigingen en stichtingen die in soortgelijke kennisorganisaties als SPIder werkzaam zijn. Ten tweede willen we ons ook gaan richten op de jongeren en dito technologie.

De komende periode gaan we deze ambitie verder concretiseren.

Tot slot wens ik iedereen voor 2010 een succesvol jaar en hoop jullie op de één van de plenaire sessies te verwelkomen.

Jeroen Macke
Voorzitter Stichting SPIder

Plenaire Sessie: Onderhoudbaarheid

Het thema van de eerste plenaire sessie in 2010 is **onderhoudbaarheid**.

Onderhoudbaarheid is een aspect van het systeem dat zich niet direct uit aan de buitenkant. Een eindgebruiker heeft hier alleen indirect mee te maken (hoe lang duurt het voor mijn probleem is opgelost?). De groei van software in allerlei producten en systemen zal ook invloed hebben op noodzakelijke capaciteit voor onderhoudswerkzaamheden.

Er zullen in deze plenaire sessie drie aspecten van onderhoud aan bod komen:

- Tim Janssen RI (Cornerstone Technology) gaat in zijn presentatie dieper in op het verbeteren van onderhoudbaarheid door re-engineering.
- Dr. Ir. Joost Visser (Software Improvement Group) gaat spreken over het meten van onderhoudbaarheid.
- Prof. Dr. Ir. Jack van Wijk behandelt het onderwerp vanuit het visualiseren van software ten behoeve van onder andere onderhoudbaarheid.

Het belooft kortom een zeer interessante plenaire sessie te worden.

We heten je graag welkom op dinsdag 30 maart 2010. We zijn dit keer te gast bij Avans Hogeschool in Den Bosch en nodigen je van harte uit om hierbij aanwezig te zijn. Meer gegevens vind je op de website van SPIder: (www.st-spider.nl).

Re-engineering legacy systems in heterogeneous IT landscape

Tim Janssen RI, Cornerstone Technology

Cost savings and reduction of complexity of the IT-legacy landscape are to day the main focus of CEO, CFO and CIO.

In the presentation for the plenary session March 30th prove will be given, that current technology is able to realize both achievements highly automated. We will show how to achieve sustainable benefits and upgrade current software assets through reduce, re-use and recycle.

Complexity and Cost reduction can be achieved in mainly two areas. In the first place through reduction of the maintenance effort, secondly through migration of expensive mainframe infrastructures toward low cost hardware and open architectures. We will demonstrate how the advanced G4-Sigma technology solution can reduce maintenance effort and migrate legacy applications to low cost Unix/Linux/Wintel based environments. The real life demo will prove that the re-engineering objectives can be achieved through total automation of both the maintenance process and migration process. In effect the automation of this IT-process itself is the key to successful projects. The project benefits are effective in the four strategic areas of maintenance, new developments, package implementation and outsourcing.

Certification of Technical Quality of Software Products

Dr. ir. Joost Visser, Software Improvement Group

The success of many organizations depends on the ability of their software systems to evolve as fast as their business processes. A system that meets its functional requirements today may not do so tomorrow if, due to low technical quality, it can only be changed with high cost and risk. The Software Improvement Group has developed a practical model for measuring technical quality of software products. In this presentation for the plenary session March 30th, Joost Visser will explain how this quality model is currently employed as the basis of software assessment, monitoring, and certification services.

Software visualisatie

(prof.dr.ir.) Jack van Wijk, TU Eindhoven

Bij de TU Eindhoven wordt onderzoek gedaan naar nieuwe methoden om abstracte gegevens in beeld te brengen. Het menselijk visueel systeem kan zeer snel patronen en structuren in complexe beelden ontdekken, bij visualisatie wordt dit gebruikt om gebruikers snel inzicht te geven in grote verzamelingen gegevens. Een belangrijk en uitdagend toepassingsgebied is visualisatie van software systemen. Deze zijn vaak groot en complex, en het ontwerpen en onderhouden ervan is een kostbare aangelegenheid. Tijdens onderhoud wordt vaak veel tijd besteed aan het verkrijgen van inzicht in de status en structuur van software. In de presentatie op de plenaire sessie van 30 maart zal een reeks voorbeelden worden gegeven hoe visualisatie kan bijdragen tot het verkrijgen van dit inzicht.

SPIDER Werkgroep Requirements

De SPIDER Werkgroep Requirements heeft een nieuwe voorzitter: Edward Crain. In dit stukje stelt Edward zichzelf voor en wat hij wil bereiken met de werkgroep.

Introductie nieuwe voorzitter

Pffft, een introductie over mezelf schrijven. Waar zal ik eens beginnen... Laat ik beginnen bij mij als persoon en daarna overgaan naar mijn ICT ervaringen, mijn relatie tot SPIDER en mijn plannen voor de Requirements werkgroep.

Persoonlijk

Ik ben 33 jaar, getrouwd en heb drie jonge kinderen in de leeftijden van 2, 4 en 6 jaar. Mijn achternaam komt uit Schotland, maar mijn ouders zijn geboren in Indonesië. Lang verhaal. Ik voel me hierdoor wel een wereldburger en mij het meest thuis in internationale teams, zoals op één van mijn vorige projecten voor een internationale

klant met collega's uit India, Engeland, België en Zweden. De cultuurverschillen, en de verschillen in tijd en locatie maken dit soort projecten complexer en uitdagender en nog meer lonend als dit leidt tot successen.

Soms ben ik op een bijna dwangneuroseachtige manier bezig met het vergaren van kennis. Zo luister ik zelfs podcasts tijdens de afwas, om ook deze tijd nuttig te besteden, lees na middernacht nog artikelen en krijg ik nog steeds verwijten van mijn vrouw dat ik tijdens het inleiden van mijn vrouw voor de geboorte van mijn jongste zoon, een boek had meegenomen over het managen van iteratieve projecten. Tja, een bevalling duurt nou eenmaal een aantal uren, kan ik net zo goed tijdens het puffen deze tijd goed benutten voor het vergaren van wat meer kennis...

Ervaring in de ICT

Ik werk al negen jaar in de ICT. Ik ben begonnen als cobol programmeur en ben snel overgestapt naar technisch ontwerper en functioneel ontwerper. Sinds 2005 werk ik als business- of informatieanalist op verschillende projecten voor verschillende klanten, hoofdzakelijk banken en verzekeraars. Ik heb met name ervaring op projecten waar sprake is van outsourcing en offshoring.

Ik heb me gespecialiseerd in alle gebieden binnen de Requirements Engineering, dus zowel Requirements Management (= tracking, tracing en change management), als Requirements Development (= elicitering, analyse, specificatie en validatie). Ik ben vooral gespecialiseerd in het schrijven van use cases in verschillende vormen en maten met de bijbehorend UML diagrammen, maar ben ook bekend met andere Requirements specificatietechnieken zoals declarative Requirements, Planguage, prototyping, personas, user stories en verschillende vormen van test driven Requirements. Naast business en informatie analyse, ben ik ook betrokken bij het maken van omvang schattingen (FPA).

Ervaring met communities

Ik speel een actieve rol binnen de business en informatie analyse practice binnen Logica NL (zeg maar de BA/IA competence unit) bestaande uit circa 300 man. Hiervoor heb ik in het verleden community wiki's opgezet, cursussen/workshops opgezet en gegeven, teamgenoten gecoached, meegeschreven aan practice team ontwikkelplannen, en heb ik allerlei hobby/werkgroepen geleid. Ik vind mijn vak leuk, en besteed naast mijn werk veel tijd aan het vergroten van mijn kennis en het actief delen van deze kennis, met name binnen mijn bedrijf Logica. Ik zal nu met mijn aantreden tot voorzitter van de Requirements werkgroep ook meer naar buiten treden.



Ik en SPlder SUGGESTIE: De link met SPlder

Via mijn Logica collega en tevens SPlder bestuurslid Martin Muller, die ik tegenkwam binnen de Logica Requirements community, en als reviewer van een aantal van mijn

functiepunt tellingen, ben ik in aanraking gekomen met SPIder. Nadat ik een aantal interne logica werkgroepen heb overdragen, en iets meer ruimte kreeg in mijn agenda, ben ik in 2008 lid geworden van de Requirements Werkgroep. Nooit spijt van gehad. Het is erg interessant om te kunnen discussiëren met vakgenoten, met een andere achtergrond, ideeën en perspectieven op het vakgebied. Dit kan ik elke Requirements professional aanraden, omdat het een completer beeld geeft over hoe het vak Requirements Engineering binnen Nederland wordt beleefd.

Voorzitterschap

Eind 2009 hoorde ik dat Arno van Herk vanwege zijn overvolle agenda het voorzitterschap van de Werkgroep Requirements wilde overdragen, en SPIder een nieuwe voorzitter voor de werkgroep zocht. Arno heeft goed werk verricht in het organiseren van de plenaire en de wat kleinere werkgroep sessies de afgelopen twee jaar. Het is dankzij hem een enthousiaste groep geworden, die nog steeds in aantal toeneemt. Uiteraard heb ik meteen aangegeven dat ik interesse heb om Arno op te volgen, en is het voor mij een grote eer dat ik gevraagd ben om de nieuwe voorzitter te worden.

Plannen met de werkgroep

Ik zal binnenkort een questionnaire rondsturen naar alle werkgroepleden, om te kijken wat de wensen zijn zodat ik de werkgroep optimaal van dienst kan zijn. Op basis van deze questionnaire zal ik komen tot een visie en strategie voor de komende jaren. Mocht bijvoorbeeld de wens zijn om ook informeel te kunnen praten met vakgenoten, dan kan ik ook bijeenkomsten organiseren in een café of op het strand tijdens een barbecue. Daarnaast zal ik een website oprichten, om andere vormen van interactie te faciliteren. Ook zal ik mijn eigen netwerk enthousiast proberen te maken om deel te nemen in deze werkgroep, om deze werkgroep in aantal te laten blijven groeien.

Nieuwe tijden

"Times are changing" zoals ik zonet op de radio hoorde van populaire Nederlandse rockgroep Di-rect. De wat jongere groep van ICT professionals lezen vrijwel niet. Ze lezen geen kranten, maar scannen het nieuws via www.nu.nl. Youtube bestaat pas vijf jaar, maar heeft de wereld volledig veranderd. Van de wat oudere vakgenoten krijg ik vaak hard copies van boeken en artikelen te leen, van leeftijdsgenoten krijg ik soft copies (=E-books en artikelen in pdf), en van de jonge vakgenoten krijg ik nu steeds vaker linkjes naar video tutorials die ze bv. onderweg in de trein op hun mobiel hebben gezien. Om deze groep ook te bereiken zal ik de mogelijkheid bieden op de website om naast de gewoonlijke blogs en fora, ook video tutorials te bekijken/up te loaden. Door kennis toegankelijker te maken met video, podcasts en bv. korte stripverhalen op deze website, hoop ik een bijdrage te leveren aan de missie van SPIder om de volwassenheid van software engineering te faciliteren.

Edward Crain (edward.crain@logica.com)
Voorzitter SPIder Requirements werkgroep

Tijd voor verandering

Zoals in deze Koerier te lezen valt, neemt Edward Crain het stokje van de werkgroep Requirements over van Arno van Herk. Edward heeft leuke plannen die toevallig (wat heet!) goed aansluiten bij de richting die Jeroen Macke met SPIder op gaat. SPIder wil

naast het bestaande ledenbestand ook een jonger publiek aanspreken. Hierbij past een andere aanpak en bijbehorende keuze van media zoals video, podcasts en andere moderne (interactieve) media uitingen. Heel veel succes Edward, we hopen snel van je te horen!

Namens het SPlder bestuur, de voormalige en huidige deelnemers van de Werkgroep Requirements bedank ik ten slotte Arno hartelijk voor zijn bijdrage in de afgelopen jaren. Omdat Requirements Engineering ook in de komende periode op de SPlder agenda blijft staan, verwachten wij dat de belangstelling voor dit interessante vakgebied groot zal blijven. Succes Arno met je verdere activiteiten en we hopen je gewoon weer als deelnemer bij SPlder events te ontmoeten!

Niek Pluijmert
Bestuurslid SPlder

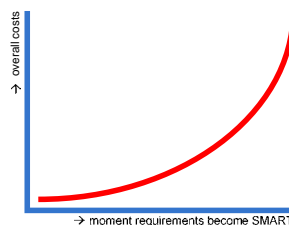
PA SMART Requirements, 100% SMART Requirements...finally!!

Changing business processes and the ICT systems they use is expensive. As Barry Boehm already concluded in the seventies, the cause for this lies in unclear and ambiguous requirements. The later requirements become clear, the higher the costs. This rise even has an exponential nature.

With the PA SMART Requirements method this problem can be eliminated. With little effort, requirements can be made 100% SMART. They can be developed and tested without question and misinterpretation. Above all, this can be done with little documentation which all stakeholders, both business and IT, can understand.

Introduction

Many books cover the topic of how to manage, engineer and specify requirements and many are quite good. One should think that no organization would have any problems whatsoever in dealing with requirements. Also one would think that developing and testing processes and systems based on these requirements would be an easy exercise. However, most know this is far from the truth. Requirements are often not as SMART as they should be. This has a negative effect on the Total Cost of Ownership of these processes and the (ICT) systems they use. The following graph (derived from Barry Boehm "The high cost of software", 1974) shows that the sooner you make your requirements SMART, the lower the overall costs.



To get SMART-ness as soon as possible you can use the method PA SMART Requirements which is based on a formal language called "PA notation". With PA notation you can define the end result of any process in a completely non-ambiguous way. Be it the end result of a business process (business requirements), a process performed by an employee (user requirements), a process which is performed by a computer (system requirements) or even deeper if needed.

Though it is a formal language, it is proven to be relative easy to learn (less than a week) and even more easy to understand (an hour on average even for most business customers). This makes it possible to use this language for both the customer as well as the developer, as well as the tester. This drives the moment requirements become SMART to a very early stage in the project, which has a major beneficial effect on the total costs.

It is the same as building a house

The principles PA notation are similar to the principles that are used in the world of constructing buildings. They mainly use a blueprint.



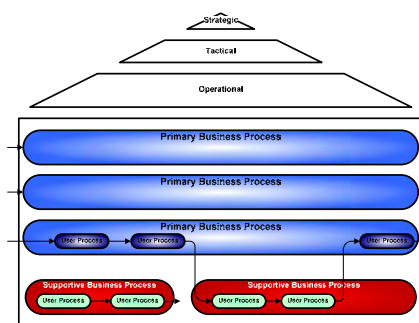
Now a blueprint works well. It should because tearing down a building after it's build is rather costly. But why does it work so well. Here are the characteristics:

- Just one interpretation. No ambiguities. Nothing left out.
- Easy to understand for all involved. Both the customer, as well as the builder.
- Short. No stacks of paper to read before you understand what should be build.
- Just “What” the end result is, never “How” this is reached.

Therefore a blueprint is something all involved can agree upon, easily. PA notation has the same characteristics. Therefore it works just as well. Also, it is based on the oldest forms of mathematics. Mainly set theory and logic.

Now what about Business, User and System Requirements

You can use these principles on organizations, on businesses. Let's start from the top. From a business.

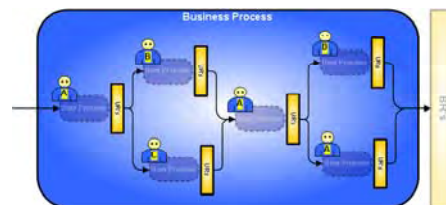


Each business has strategic, tactical, operational steering and processes. Lots of processes. Now when using PA notation we initially are not looking at how the process looks like, but what it should result in.



So in this case we only look at the end result of a business process. Within PA SMART Requirements, the end result of a business process and the rules that apply to it, are called Business Requirements (BR's). With PA notation you can specify them completely SMART. All the details you need to start developing and test a business process that will achieve the specified end result are provided with PA notation. Nothing more, nothing less. The language even forces you to give just the right amount of details.

When the end result is clearly specified, or a part of the end result (if you're working in iterations), you can start to develop the process. In PA SMART Requirements this is easy. With a method called Backward Chaining¹, you can derive the most optimal business process (tasks). In other words the most optimal way the human actors should perform their individual user tasks and in which order.



In the same paradigm, PA notation can be used to specify the results of each individual user process and the rules that apply to it fully SMART. In PA SMART Requirements, these are called User Requirements (UR's). So if commissioned to develop and test a single user process, you've already got exactly the details you need to start developing and testing it.

Of course when zooming in one step further you could derive the way the most optimal user process looks like. In other words the most optimal way the system actors should perform their individual system processes and in which order. These system processes could interact with humans actors (users) or be fully automated.



Whichever the case, we can use the same paradigm again on these system processes as we did before for user processes and business processes. In other words PA notation can be used to specify the results of each individual system process and the rules that apply to it fully SMART. In PA SMART Requirements these are called the System Requirements (SR's). So if commissioned to develop and test a single system process, you've already got exactly the details you need to start developing and testing.

¹ http://en.wikipedia.org/wiki/Backward_chaining

Mapping it all to the V-model

Now PA SMART Requirements does not force you to use just one development method. You can use waterfall approaches (e.g. SDM) as well as iterative approaches (e.g. RUP, Agile, Scrum). But many development methods have the same characteristic approach. To clarify how you can use PA notation within a development cycle we use the well known V-model to show where PA notation is used.



For simplicity we've omitted system architecture and technical design, but when we look at the V-model you can conclude that PA notation can be used from the business level (business requirements for business processes) until the system level (system requirements for system processes). As said before more decomposition levels can be introduced if need be, or even less if possible.

When starting on coding (or technical design if this is necessary), requirements specified with PA notation are used to develop the process. The goal is to develop a process which achieves the specified end result of that process.

Of course, the left side of the V-model is all about testing if the process actually achieves the specified end result. There the same requirement specifications (written in PA notation) are used.

Main advantages of PA SMART Requirements

For some readers the advantages of using PA SMART Requirements are clear already. But just to be on the safe side, the main advantages are listed here:

- **Can be used as early as Business requirements:**
PA SMART Requirements is not a method that can be used only for developing system processes. With PA notation also Business Requirements can be made fully SMART so that they can be developed and tested without question.
- **One language for both Business and IT (optimal Business & IT alignment):**
Just one language is used during the complete development cycle. Understandable for both business and IT, this results in an optimal business & IT alignment. Of course this also makes it unnecessary to create several translations from something the customer understands, to something the developer needs to start developing and the tester needs to start testing.
- **High quality requirements, specs and designs (short, simple, comprehensible):**
If you use PA notation for your requirements, specifications and designs, you get fully understandable ones. Also all involved can have only one interpretation of what the end result should be. This results in requirements, specifications and designs of a much higher quality than one is used to.
- **SMART requirements, specs and designs can be developed and tested directly:**
No questions need to be asked about these requirements, specifications and designs. Therefore development and testing (test preparation) can start right away. If used properly there is even no need for reviewing. At least not to check if it can be developed and tested.
- **Anywhere in the world (optimizes Offshore development):**
Of course something that can be interpreted in only one way, will be interpreted

in this one way anywhere in the world. This optimizes offshore development and testing.

- **Makes 95% of all (IT) designs redundant:**
Using just one language makes it possible to define a limited set of design patterns that will be used to develop the process and its ICT systems. In most cases this makes technical designs unnecessary.
- **Reduces (FTE) cost of development and Test:**
If it is 100% clear what the system should be able to achieve as the end result, then the communication that is needed to get this clear understanding is limited to almost nothing. Also substantially less errors and redesign is necessary to develop and test. This reduces development and testing costs drastically.
- **Less software defects:**
As a result of the above, there are much less software defects. Most noticeably is the reduction of defects at the very first test run.
- **Fits within existing methods:**
The nice thing about PA SMART Requirements and PA notation is that it can be added quite easily to existing methods. It has already been applied successfully in methods such as RUP, (D)SDM, Agile, Scrum, Prince2 and some customer specific methods. It can be seen as a small but crucial ingredient which can be added without changing the overall method. If a customer doesn't have an existing method, one can always adopt the one that suits his needs the most. If needed, PA SMART Requirements is equipped with an effective and structured approach for business processes and ICT development.

An example "Billing with approval"

In this chapter we'll discuss an example of a business process which is all about employees that are hired by one or more customers which will be billed for the time the employees spent on tasks. In this case we'll only show the parts that concern hours spent on task, approval of those hours and some management information that will be sent directly to the manager of those employees.

Here follows (in blue) the specifications of the end result of this business process mixed with explanations (in normal).

For each DAY in "FIRST DAYS OF THE MONTH" the following applies:

"Hours are entered and approved" **and**

"Managers are informed"

"FIRST DAYS OF THE MONTH" =

TIMESTAMPS(day = 1 **and** time = 0:00 **and** timestamp < NOW)

"Hours are entered and approved" =

For each E in EMPLOYEES the following applies:

One T exists in TIME_REGISTRATIONS with:

year, month = **input from** EMPLOYEE

OF_EMPLOYEE = E

approved = **input from** CUSTOMER **based on** [T.*, L.*]

and

Several L exist in REGISTRATION_LINES with:

day, task, location, hours = **input from** EMPLOYEE

TIME_REGISTRATION = T

```

“Managers are informed” =
  For each MANAGER in EMPLOYEES(is_manager) the following applies:
    One MAIL exists in MAILS with:
      from   = ‘noreply@logica.com’
      to     = MANAGER.e_mail_adres
      subject = ‘Registration overview: in total <sum(L.hours)>
registered’
      body   = “message content”

“message content” =
  <<RTF>>

“Hour overview” =
  <<RTF>>

“Hour list” =
  For each TM in MANAGER.TEAMMEMBER.sorted_by(last_name):
    <<RTF>>

“Approved Yes/No” =
  If TM.TIME_REGISTRATION(“disapproved”) exist
  then ‘No’
  else ‘Yes’

“disapproved” =
  year = NOW.year and
  month = NOW.month and
  not approved

```

Edwin Hendriks | Logica Business Consultant (Process and ICT improvement)
Edwin.Hendriks@Logica.com

SPIDER Werkgroep P-CMM

Ben Linders, Adviseur

SPIDER houdt zich meestal bezig met thema's als software processen, test methodes, etc. Maar om succesvol te zijn met softwareontwikkeling is het ook noodzakelijk om een goede en gemotiveerde workforce te hebben. Het SEI heeft het People Capability Maturity Model (P-CMM) model ontwikkeld om dit systematisch te kunnen bereiken. Het P-CMM is een soortgelijk model als andere CMM's. Het kent vijf maturity levels en een aantal Process Areas zoals Compensation, Coaching, en Performance Management. Het biedt dezelfde voordelen die we kennen van CMMI: een heldere structuur, ingebouwde prioriteitsstellingen, gedetailleerd uitgewerkte process areas, een assessment methode, etc.

De P-CMM werkgroep heeft als focus de menselijke aspecten die een rol spelen bij Software Process Improvement en Kwaliteit. Als leidraad gebruiken we het P-CMM model van het SEI. Maar alle ervaringen waarin de focus op de mens ligt zijn welkom,

ook als ze niet vanuit het P-CMM geïnitieerd zijn, we zoeken dan wel naar de relatie met het P-CMM omdat we dat model als gemeenschappelijk framework en "taal" gebruiken.

De werkgroep richt zich op drie aspecten:

- Het model.
- De praktijk.
- Business Case en Marketing van P-CMM.

In iedere bijeenkomst wordt een thema behandeld, waarbij we bovenstaande aspecten gebruiken om dat thema van verschillende kanten te belichten. Het doel is om van elkaar te leren; brengen en halen.

In 2010 zijn de volgende bijeenkomsten gepland:

- 15 maart: Workforce Planning.
- 31 mei: Competence Development.
- 23 augustus: Career Development.
- 11 oktober: Competency Based Practices.
- 13 december: Workgroup Development.

Voor meer informatie over de werkgroep, zie:

- P-CMM discussie platform: <http://www.workforcematurity.nl/>
- LinkedIn groep: <http://www.linkedin.com/groups?home=&gid=1939585>
- SPIder werkgroep pagina: <http://www.st-spider.nl/WG/WGPeopleCMM.html>

De trekkers van de werkgroep zijn Cecile Davis (praktijk), Herman van Dellen (model), Andre Heijstek (algemeen) en Ben Linders (Business Case/Marketing).

Deelname aan de werkgroep is kosteloos. Dus, heb je interesse meer te leren over de menselijke kant van SPI en kwaliteit, neem dan contact met ons op. Je bent van harte welkom bij de werkgroep!

Nieuws van de Q Society

De Q-Society, voluit de Society voor Quality Professionals in ICT, gaat ook in 2010 volop door. Ook dit jaar willen we weer twee evenementen organiseren. Voor het voorjaarsevenement zijn de voorbereidingen gestart: er is een groepje geformeerd die de organisatie op zich neemt. Thema en datum zijn vastgesteld. Als werktitel hebben we gekozen voor:

Kwaliteit in de cloud

Wat betekenen moderne ontwikkelingen als SAAS en Cloud Computing voor de kwaliteitsmanager?

Verandert er iets voor de kwaliteitsmanager?

Wat betekenen deze ontwikkelingen voor het kwaliteitsvak?

Wil je het antwoord op deze en andere vragen, houd dan de morgen van 8 juni vrij in je agenda. We zijn dan te gast bij de TU Eindhoven.

Na een aantal aansprekende presentaties, zal de afsluitende spreker ons letterlijk in de wolken brengen....

Tot ziens op 8 juni!

Niek Pluijmert
Bestuur SPIder

Harvesting service orientation

Wil Leeuwis, Rabobank

In order to get optimal results, it's worthwhile to look back every now and then, and ask the question: "What did we learn?" In our days of Red Queen-talk, the paradox is that effective learning, and thus taking some time every now and then in order to harvest from past seeding, is more necessary than ever to get on-time quality results. Today's conundrum is the same as it has been for long, today's answer is: services. So I'll point my arrows at services.

Models and systems

Models are simplified representations of everyday reality. Systems are a special kind of models in which the universe or a part of it, is modeled in the form of elements and relations between those elements. Within systems we can define subsystems, subsets of elements sharing certain characteristics, and aspect-systems, subsets of the relations sharing certain characteristics. On top of that the whole thing is recursive: a set of systems is a system in his own right. Founded in the 1940's, systems theory offers a vast amount of conceptual groundwork, ready to aid twenty-first century problem analysts and solvers in their quest to get insight in the complexities under investigation [1], [2], [3].

The reason we construct models and systems is to get a better understanding of certain aspects and parts of the universe around us or within us. That understanding is gained at the expense of insight in and understanding of the parts and aspects we left out, we abstracted from. Choosing the scope of our model follows from our objectives, from what we want to learn or to communicate: the making of models is a purposeful undertaking.

Mathematical models differ in an important aspect from other models: mathematics create, by defining axioms, a universe of their own. Within a mathematical model things are perfectly clear, even when doing fuzzy logic. In the set of natural numbers, every number is either odd or even, and there is not one even number that is more or less even than any other even number. How different from the universe as we know it!

Everything is vague

In the real world, everything is vague to a degree you do not realize till you have tried to make it precise, and everything precise is so remote from everything that we normally think, that you cannot for a moment suppose that is what we really mean when we say what we think [4].

Being so familiar with the mathematical abstractions we are used working with, we tend to forget this. And when we forget it, we mistake our model for the real world. Mathematics are great. Thanks to mathematics lots of real world problems are solved. But every engineer, enthusiastically using mathematical models of the real world, knows the model is just that, the model. Because the real thing has its equivocalities, its aberrations from what, in terms of the model, is normal. It is this two-facedness of models that led one of my teachers to the maxim "Don't believe the model. Don't ignore the model". Don't mistake the model for the real thing, be aware that things are left out of the model: surprise ahead. Having said that: use the model if every way you can. It will help you survive in a world that's too complex for a mortal brain.

Models adding to complexity?

Models, once described, become a part of the universe. That's kind of counterintuitive and unwelcome: starting out to make a model with the intention to get a grip on

complexity, we end up with a universe even more complex. There are three things to be said about this. First: It all depends on your viewpoint. From an overall viewpoint it's true that complexity is raised. But from a local viewpoint a well-formed model will surely help in getting a grip on complexity. It's an example of the law that there is no such thing as a free lunch: the cost of the local gain is a global loss-just as it often is. Second: Adding the model to the universe bring forth ambiguity. For the car-object instantiated from the car-class certainly differs from the car coming from the GM production line. So we should carefully choose our words, when both the real world and the model are in the scope of the conversation we're having. Third: Professionals have a responsibility to the company they're working for, and to society as a whole: they should not add to complexity without a very good reason.

Services

Services, service orientation, service oriented architectures. Something old, something new, something borrowed, something blue. Old wine or real innovation? In my opinion: both. There is a lot to be gained if we stay away from the either-or fallacy in the services area. Instead of searching for arguments that it's CORBA revisited or what have you, think broader. Because there is a lot of old, well understood and practically applied theory that can help us harvesting the profits of the innovation part of the services-world.

Just for the sake of this article, let's build a model to get a firmer grip on the notion "service". We'll undo the model at the end of the article, so there will be no harm done to overall complexity. The model I propose consists of four components: a memory, a processor, a connection between the memory and the processor, and an actuator. The memory represents all the internal and machine readable memory in the universe, and the same goes, m.m., for the processor and the connection, the last one thus including things like the internet. The actuator stands for all the machines and human beings in the world that, for the sake of simplicity, can interact directly with the memory, i.e.: they can read from and write to memory. Since the combination of processor, connection and memory is a machine, the model can be used to describe recursive processes.

Gaining some insight from our model

I leave it to the reader to do some detail design on this model. You might, for example, position executable programs in memory or in the processor, with certain consequences for the description of the execution process. But even in the loosely described model, it's clear that a call to a subroutine in the 1950's assembler era is identical to a PERFORM in time-enduring Cobol and to a web service in the 2008 Web 2.0+ era.

What is happening? What is the process? There is optionally some data gathered. Then there is a request to a different part of the universe to do something, often with the gathered data supplied as parameters. This is followed, either directly or after some time has passed, by execution of code - as part of which a request might be sent to yet another part of the universe, and so on. When execution is done, a message might be send to the requester that the works done, and this might be accompanied by some data.

So what's up? My first observation is that in all this, you can't point your finger at the service. In all the currents flowing in our simple model, it's hard to point out what that service really is. To a large degree it's the name by which we know it. For the rest there are process steps: calling, executing code, changing the values stored in certain memory locations, taking action when a value in a certain memory location gets or exceeds a certain value, and so on. Might we conclude that a service is a reification,

that by naming it we have given thing-like existence to an abstraction? Or that a service is an emergent property, an epiphenomenon, coming into existence because all of these activities happening? I don't think playing with words has much value, so I guess we should be happy with the ability to name a service, no further questions asked.

Now what has changed since the branch to subroutine-era? Since Darwin we know there are no essentials in the real world, but IT systems come so close to their mathematical models that within certain constraints it's fair to ask the question: What essential differences are there in contemporary applications, when compared to their legacy counterparts, that makes the services notion worthwhile? The answer to this question is simple: complexity. In the assembler-days a cross-reference listing was about all you needed to have an overview of your, well, services and where and by whom they were used. In the 2008 web service version of the story, you really need models and systems to guide the development and use of services. Services are complex on all levels. If you need examples of devils hiding in details, go investigating service orientation being put to practical use. Making services available in a form that can be used by the business to create business processes using those services is complex, it requires service directories that state the goal of the service in business speak. At the technical level there is a lot of complexity following from all the different platform and network technologies that have to seamlessly work together in a services environment. At the design and build level these business oriented and technology oriented elements are to be woven together. Once deployed, service levels must be met. And all this in a world that just loves a daily dose of change in all those areas. So building to last is no good, services have to be built to change. And I did not even dare to mention information security!

This would all be quite irrelevant if services were of marginal interest. In fact, it's just the opposite. They bring with them the promise of agility so your business loves them! So, in order to move fast, we have to stand on giants' shoulders. The notions of coupling and cohesion, structured analysis and design, encapsulations, the use of patterns, it's all there for the taking [5], [6], [7].

Words

Words that are used in combination with service orientation are, among others: architecture, maturity model, roadmap and governance. Is there any complexity reduction to be done in this area? I think there is.

Maturity models are about best practices. That means they are about real, existing processes which, under certain conditions, yield high quality results. Maturity models can be used to guide organizations to reach higher performance levels by directing their attention to process areas that have proven to be of importance. A maturity model used in this way acts as a roadmap: it guides you on the trip from here to there. But another kind of roadmaps is out there, in which the destination is not known in the sense that we can point our finger to a real existence of the so called destination. Instead of that, the destination is a situation that is perceivable, and declared desirable. The next step in a situation like that is a planning process to fill the gap between the situation at hand and Nirvana by means of intermediate steps. Nothing wrong with this kind of migration, on the contrary. But what should not be done, is calling the resulting model a maturity model. When you do that, you're stretching the meaning of the notion maturity model to an extent that makes it meaningless. When it is done this way, there's only global complexity added, without a countervailing local reduction in complexity.

Is service orientation something completely different? From certain perspectives, certainly. Does it need its own maturity model, governance and so forth, just based on its being different? That depends. The development of a comprehensive maturity model for something as broad as service orientation, potentially ranging in scope from business processes to XML and semantic interfaces, is not a trivial task. If possible, we should use elements readily available, to get such a maturity model ready in time for planned use. Applications as far apart as Photoshop and Eclipse have the same mechanism for doing just that: the plug-in. Especially in a service oriented context, evaluating existing models for reuse should be the first step. When taking the plug-in perspective, lots of models like CMMI, ITIL and COBIT can contribute to service orientation implementation speed and business success.

So, once more, "keep it simple" is the maxim. Don't add models unless not adding them would be worse, adhering to Einstein's plea to express everything as simple as possible, but no simpler than that. Stick to the established meaning of words: maturity models are about best practices, roadmaps are used to get to known destinations in a Rand McNally-fashion, and migration plans are for reaching Nirvana. And last but not least: reuse, reuse, reuse.

Last word on words: ABC

There is an infamous trio known as ABC, academics, business and consultants. The academics forging new theories. The consultants, translating these in their own words to palatable piecemeal for the business. And the business, paying the consultants, and giving the academics the opportunity to wander around in their organizations, gathering material to base new theories on. It is clear that, in order to earn some money in this area, you should not use words with a clear given meaning. Instead of that, you play the redefinition-game. In marketing-speak: you differentiate. But with the same effect: lots of word games, less work done. Could a process like this one explain why we see so many new models, and so few initiatives to make models fit for change - or even fit for use - and to keep them up to date?

End of file

Keep in mind that models are there to abstract from aspects or parts that are of less concern to you and your objectives. The models that are made to communicate a consultant's story is not necessarily helping in solving your problem. As Mark Anthony Luhrmann said: "Be careful whose advice you buy, but, be patient with those who supply it. Advice is a form of nostalgia, dispensing it is a way of fishing the past from the disposal, wiping it off, painting over the ugly parts and recycling it for more than it's worth."

If this article were a service, the request might be: "How can I harvest service orientation", and the data sent back would read: "Try to make clear what's new. Invest in mastering the new. Don't forget that there is a lot you, and others before you, have learnt about things more or less like it. Reuse that. The new parts are complex. Reuse is hard. Let the force be with you."

1] Kenneth E. Boulding, 1956. General Systems Theory, The Skeleton of Science. In: Management Science, 2, 3 (Apr. 1956) pp.197-208; reprinted in General Systems, Yearbook of the Society for General Systems Research, vol. 1, 1956.

[2] W. Ross Ashby, 1956. An Introduction to Cybernetics, London, Chapman & Hall. Available on the internet at: <http://pespmc1.vub.ac.be/books/IntroCyb.pdf>.

[3] Ludwig von Bertalanffy, 1968. General System Theory: Foundations, Development, Applications, New York: George Braziller, revised edition 1976: ISBN-13: 978-0807604533.

- [4] Bertrand Russell, 1918. Lecture: The Philosophy of Logical Atomism. Bertrand Russell, David Peers (ed.), 1985. The Philosophy of Logical Atomism, Open Court Classics. ISBN-13: 978-0875484433. Also available via the Amazon Online Reader.
- [5] Edward Yourdon and Larry L. Constantine, 1979. Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Prentice Hall. ISBN-13: 978-0138544713 facsimile edition 1986.
- [6] Glenford Myers, 1979. Reliable Software Through Composite Design. Van Nostrand Reinhold, ISBN-13: 978-0442256203.
- [7] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, a.k.a The Gang of Four, 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN-13: 978-0201633610.

The concept of services is important to business and to society in general. It means a step forward to the boundless flow of information, the vision of The Open Group, and as such also to the universal access to all human knowledge as envisioned by Brewster Kahle's Internet Archive. Both visions are inspiring!

De kleine CMMI trilogie compleet

Jan Jaap Cannegieter – Adjunct Directeur SYSQA B.V.

Rini van Solingen – Docent Technische Universiteit Delft en zelfstandig adviseur.

In 2006 is 'De kleine CMMI' uitgebracht met als doel om de Nederlandse professional te helpen met een kort en bondig, Nederlandstalig overzicht van CMMI voor ontwikkeling. Niet voor de specialist, die maakt doorgaans wel gebruik van de originele stukken van het Software Engineering Institute (SEI). Maar juist voor iedereen die geen expert wil zijn, maar wel in aanraking komt met CMMI in een prestatieverbetering traject. Inmiddels is het boekje in de derde druk en aangepast aan recente versies van CMMI en daarnaast zijn verbeter suggesties van lezers verwerkt.

Het SEI heeft besloten om naast een CMMI voor ontwikkeling ook een CMMI voor acquisitie en een CMMI voor diensten uit te brengen. CMMI voor acquisitie is een volwassenheidsraamwerk voor het beheersen van uitbesteding en aanschaf van producten en diensten. Populair gezegd is CMMI voor *acquisitie* een volwassenheidsmodel voor opdrachtgevers. CMMI voor *diensten* is een volwassenheidsraamwerk voor het beheerst ontwikkelen en leveren van diensten. Je kunt hierbij denken aan een beheer afdeling van een IT organisatie, maar CMMI voor diensten richt zich juist ook nadrukkelijk op organisaties die diensten leveren buiten de IT. Bijvoorbeeld een supermarkt, een transportbedrijf of een uitzendbureau. CMMI voor acquisitie bevat 16 procesgebieden die ook in CMMI voor ontwikkeling zitten: de zogenaamde kern procesgebieden. Daarnaast kent CMMI voor acquisitie zes specifieke acquisitie procesgebieden. Naast diezelfde 16 kern procesgebieden kent CMMI voor diensten zeven specifieke dienst procesgebieden.

Het vrijgeven van CMMI voor acquisitie heeft in 2009 geleid tot de uitgifte van 'De kleine CMMI voor acquisitie'. Binnenkort verschijnt nu ook 'De kleine CMMI voor diensten'. De doelstelling van deze 'kleintjes' is hetzelfde als van 'De kleine CMMI voor ontwikkeling': het geven van een kort en bondig, Nederlandstalig overzicht van het betreffende CMMI voor de Nederlandse professional. Elk van deze boekjes bestaat uit een beschrijving van het model, een beschrijving van de implementatiemethode IDEAL en een uitleg van de werking van CMMI onderzoeken (assessments). Daarnaast is er een beschrijving van alle specifieke procesgebieden zodat er een duidelijk inzicht wordt gegeven in de waarde van de betreffende CMMI variant. Ook zijn vertalingen van alle doelen en praktijken opgenomen.

Moet je deze boekjes nu alle drie kopen? Als auteurs zeggen we natuurlijk volmondig “ja!”. Tegelijkertijd is het aan te bevelen juist die variant te kiezen die in de eigen praktijk wordt toegepast. Als het primaire proces dienstverlening is, is ‘De kleine CMMI voor diensten’ geschikt voor jou. Ben jij vooral betrokken bij uitbesteding of aanschaf van producten of diensten, dan is ‘De kleine CMMI voor acquisitie’ geschikt voor jou. Organisaties die al bezig zijn met CMMI voor ontwikkeling en hun opdrachtgeverschap en/of servicemanagement willen verbeteren hebben zeker wat aan deze andere CMMI varianten omdat ze iets meer detail bevatten en dus gericht hulp kunnen bieden.

Publicatie CMMI voor Ontwikkeling

Lammert Braafhart – CMMI expert en management consultant bij Logica.

Het gebruik van CMMI modellen als standaard voor procesverbetering is wereldwijd enorm toegenomen. In veel landen bestaat de behoefte “best practices” in de eigen taal te kunnen lezen, om de essentie ervan zo goed mogelijk te kunnen begrijpen en te kunnen toepassen. Gebleken is dat ook in ons land deze behoefte bestaat. Diverse landen zijn ons al voorgegaan met het uitbrengen van een vertaling van CMMI, zoals Japan en China maar ook Spanje, Frankrijk en Duitsland. En er zijn diverse andere landen die eveneens plannen hebben met het uitbrengen van een vertaling. Alle Europese vertalingen zijn gepubliceerd door Pearson Education. Al een paar jaar geleden startte ondergetekende met het vertalen van CMMI for Development. Deze versie is nu in concept gereed. Voor de Nederlandse vertaling zijn de eerste verkennende gesprekken met Pearson Education Benelux gevoerd.

De oorspronkelijke plannen voor de Nederlandse vertaling richtten zich op het uitbrengen van versie 1.2 van het CMMI voor Ontwikkeling. Het Software Engineering Institute (SEI) heeft echter voor november 2010 de publicatie van versie 1.3 van haar CMMI modellen gepland. Dit is aanleiding om de plannen voor de publicatie van de Nederlandse versie hierop aan te passen. Wij richtten ons nu op versie 1.3 van het “CMMI for Development”.

Op dit moment is de vertaling van versie 1.2 rond (gereed voor review). Op het moment dat de tekst van versie 1.3 wordt vrijgegeven zal ook de vertaling daarop worden aangepast en geverifieerd. Zodra zich voldoende gekwalificeerde reviewers hebben aangemeld, kan de door het SEI vereiste onafhankelijke verificatie starten. Dat doen we op de versie 1.2 welke activiteit hervat zal worden na de launch door het SEI van versie 1.3. Conform informatie van het SEI wordt verwacht dat het aantal aanpassingen niet al te groot en ook niet echt ingrijpend zullen zijn. Zodra er een gereviewde versie 1.2 ligt verwachten we die snel om te kunnen zetten naar versie 1.3. Overigens is het SEI voornemens om de CMMI modellen voor acquisition, services en development allemaal in een keer aan te passen en consistent te maken en naar versie 1.3 te tillen. In het artikel van Jan Jaap Cannegieter wordt nader ingegaan op de plannen voor de reeks van de kleine CMMI.

Het reviewen van CMMI voor ontwikkeling zal onder auspiciën van SPIDER worden uitgevoerd. Martin Muller zal namens het SPIDER bestuur de coördinatie op zich nemen. Zoals het van het SEI kan worden verwacht wordt een Verificatieplan geschreven en getoetst, heeft het SEI eisen opgesteld waaraan reviewers moeten voldoen, zoals aantoonbare kennis en ervaring met CMMI en wil het SEI zelfs na afloop de review findings van SPIDER ontvangen! Dit allemaal natuurlijk om de kwaliteit van de vertaling zo hoog mogelijk te maken (kwaliteitswaarborg SEI).

Na goedkeuring van de Nederlandse vertaling door het SEI zal toestemming worden gegeven voor publicatie. Dit zal naar alle waarschijnlijkheid nog voor het einde van dit jaar plaatsvinden. Wij zullen u op de hoogte houden van de verdere ontwikkelingen.

Heeft u kennis en ervaring met CMMI in de praktijk en heeft u interesse, tijd en gelegenheid om als reviewer een bijdrage te leveren, dan kunt u een mail uitsturen naar ondergetekende. U kunt anders ook een mail sturen naar Martin.Muller@Logica.com. Alvast bedankt voor uw getoonde belangstelling en mogelijke bijdrage.

Lammert Braafhart

Lammert.Braafhart@Logica.com

De SPIder Organisatie

SPIder is de Nederlandse netwerkorganisatie voor SPI. SPIder organiseert jaarlijks een conferentie, minstens drie plenaire sessies met sprekers met variërende thema's. In aparte werkgroepen worden thema's bediscussieerd en verder uitgewerkt. De SPIder koerier is het medium dat minstens viermaal per jaar verschijnt. Hierin kunnen lezers hun mening uiteenzetten en interessante ervaringen delen met SPIder leden. SPIder is een stichting, een non-profit organisatie die wordt bestuurd door vrijwilligers.

Lidmaatschap van SPIder is gratis, en alle activiteiten met uitzondering van de jaarlijkse SPI conferentie zijn gratis toegankelijk. Dit wordt mede mogelijk gemaakt door onze reguliere sponsors. Dus dank voor de bijdrage van Philips, Sogeti, DNV-Cibit en SIG.

Ook kent SPIder donateurs, zowel bedrijfsmatig als individueel. Donateurs hebben bij ons een streepje voor, en krijgen extra voordeel op de activiteiten van SPIder en van onze zusterverenigingen. Wil je SPIder ook steunen, meld je dan aan bij het SPIder secretariaat.

Het SPIder bestuur bestaat uit de volgende personen:

- Jeroen Macke, voorzitter
- Niek Pluijmert, penningmeester, Q Society
- Martin Muller, strategie, donateurs
- Kasia Wiacek, plenaire sessies
- Wil Leeuwis, Website, SPIder conferentie
- René Krikhaar, SPIder conferentie, SPIder Koerier

Informatie over SPIder is te vinden op de website: www.st-SPIder.nl

Voor reacties en bijdragen op de **SPIder website** kunt u zich richten tot:

Redactie SPIder web, Wil Leeuwis

E-mail: w.leeuwis@gmail.com

Opstarten nieuwe Werkgroep SPIder

Nieuw bloed

Het SPIder bestuur hecht grote waarde aan laagdrempeligheid voor de SPIder community, om in staat te zijn kennis en ervaring zo goed en zo snel mogelijk met elkaar te kunnen en blijven delen. We moeten daarbij uiteraard waken voor de rechten van eigenaars van artikelen, presentaties, white papers en andere bijdragen. Alle materialen zoals opgeleverd in de verschillende werkgroepen moeten in principe beschikbaar worden gesteld voor de hele SPIder community. Dit uitgangspunt geldt mogelijk niet altijd; zo zal materiaal dat als ingangsdokumentatie gebruikt wordt naar beëindiging van de werkgroep activiteiten opgeruimd worden. Het is gebruikelijk bij SPIder werkgroepen dat deelnemers van een werkgroep speciale toegang krijgen tot de work space van de werkgroep. Binnen de werkgroep zijn producten als het ware vrij

verhandelbaar en worden producten aangemaakt/ontwikkeld. Het is dus voor een doorsnee SPIder lid niet mogelijk om van buiten in de keuken van een werkgroep te kijken. Wil je dat toch, trek dan een schort aan en doe mee!

Wergroepen en SPIder Koerier

Het SPIder bestuur faciliteert werkgroepen. Ieder bestuurslid van SPIder heeft een of meerdere werkgroepen onder zijn of haar hoede. Het vraagt de werkgroep voorzitters periodiek de status m.b.t. de werkgroep in de SPIder Koerier toe te lichten. En voor de werkgroepen die een concreet product nastreven vragen wij om acte de presence te geven op een van de goed gewaardeerde en bezochte plenaire sessies.

Doelstellingen

De doelstellingen van werkgroepen kunnen best uit elkaar liggen. Dit hangt af van zaken als: ontstaansgeschiedenis, type en onderwerp van de werkgroep, omvang en persoonlijke voorkeur van de werkgroepleden. Zo zijn er werkgroepen die concreet gericht waren op ontwikkeling en oplevering van een bepaalde deliverable, CMM Starterkit, Modellenboek en recentelijk: SEI Technical Document on CMMI road maps (werkgroep CMMI Roadmaps) en het SPIder vakantie veranderboek (werkgroep Invoering), om zich daarna op te heffen of een doorstart te maken en een volgende deliverable op te bouwen. Maar een werkgroep kan ook gericht zijn op alleen het netwerken en kennisdeling zonder concreet een deliverable te willen maken. Alles kan en niets moet bij SPIder.

Het bestuur ondersteunt nieuwe initiatieven van harte. Zo hebben we naar onze sponsors gecommuniceerd dat we vanaf 2010 elk jaar een nieuwe werkgroep zullen starten. Uitdagend? Ja. Haalbaar? Met jullie inzet en medewerking zeker!

We zien belangstellenden die een nieuwe werkgroep willen opstarten graag tegemoet. Een belletje of mail naar SPIder voldoet. Wij helpen je een werkgroep vorm te geven. Alle suggesties zijn meer dan welkom!

Martin Muller
Bestuurslid SPIder

Colofon

De inhoud van de SPIder Koerier wordt verzorgd door het SPIder netwerk. Dat betekent dat de redactie met plezier artikelen ontvangt voor publicatie. Goede ideeën om op andere wijze zinvolle inhoud te geven aan de Koerier zijn welkom: een speciale reeks van artikelen rondom een bepaald thema, een discussie via de Koerier in de vorm van meningen en opinies of een column. Uw ideeën zijn welkom, we kunnen ze bespreken tijdens de conferentie, per mail of telefoon.

De SPIder Koerier wordt gemaakt onder auspiciën van SPIder door René Krikhaar en Cantrijn Secretariaten. Voor artikelen, mededelingen, reacties en vragen m.b.t. de **SPIder Koerier** kunt u zich wenden tot de schrijvers of

SPIder Koerier

E-mail: koerier@st-SPIder.nl

Volgende deadline van de SPIder koerier is 15 juni 2010.

Nieuwsberichten & evenementenkalender

De evenementenkalender bevat een overzicht van nationale en internationale conferenties, seminars en soortgelijke activiteiten op het gebied van SPI en kwaliteit, maar ook op aangrenzende gebieden die van interesse kunnen zijn voor de SPIDER deelnemer. Zoals sessies georganiseerd door Agile Holland, Testnet, ASL-BiSL Foundation, ITSMF, INCOSE, PMI, etc.. Daarnaast zijn de activiteiten van SPIDER opgenomen.

Via de SPIDER Koerier kan een organisator van SPI gerelateerde evenementen een selecte groep van geïnteresseerden bereiken.

Ⓢ = SPIDER event

✓ = korting voor SPIDER donateurs

	2010		
	23 maart	NAF mini conferentie Agile en Architectuur http://www.naf.nl/	
	23 maart	GPI Consultancy IT-SEMINAR: INTERNATIONALISERING EN GLOBAL SOURCING www.gpic.nl/seminar.htm	
Ⓢ	30 maart	SPIDER plenaire sessie "inzicht in uitbreidbaarheid", Avans Hogeschool Den Bosch www.st-spider.nl	
Ⓢ	8 april	Synergio: TopTeam – De business analyst workbench	✓
	21 april	Dutch Testing Conference, Spant Bussum http://www.dutchtestingconference.nl/	✓
	28 juni-1 juli	SEPG Conference Porto Portugal http://www.sei.cmu.edu/sepg/europe/2010/	✓
	25-26 oktober	ITSMF jaarcongres http://www.itsmf.nl/	